References

[1] D. Simpson, Anal. Proc., 21 (1984) 385.
[2] D. Simpson and W.G. Simpson, An Introduction to Applications of Light Microscopy in Analysis, Royal Society of Chemistry, London, 1988, p. 164.
[3] D. Simpson and W.G. Simpson (Editors), The COSHH Regulations: A Practical Guide, Royal Society of Chemistry, Cambridge, 1991, p. 1.
[4] D. Simpson, Chromatography, 39 (Oct./Nov.) (1995) 9.

Dr. Diana Simpson is at Analysis For Industry, Factories 2/3, Bosworth House, High Street, Thorpe-le-Soken, Essex CO16 0EA, UK. She has over 30 years experience as an industrial analytical chemist in various fields covering food, water, plastics, pharmaceuticals and occupational hygiene. The practice of Analysis For Industry was established in 1975 with Dr. Simpson and her husband as Principal Consultants and she has been involved in forensic science and expert witness work since that time.

computer corner

Using dynamic data exchange to exchange information between Visual Basic and Matlab

Application to a diode array spectrophotometer

Eduardo O. de Cerqueira,
Ronei J. Poppi*
Campinas, Brazil

Photodiode array spectrometers are increasingly being coupled to flow analysis and separation techniques. Computer programs are needed to control the data acquisition from each photodiode and to handle the data. The Microsoft Visual Basic programming language is used along with a DLL library and dynamic data exchange (DDE) to allow communication with the MATLAB computational environment.

1. Introduction

Photodiode array spectrophotometers are increasingly being used in analytical chemistry because they contain no moving parts and a full spectrum can be obtained in a few milliseconds [1], making possible their application with techniques such as high performance liquid chromatography [2] and flow injection analysis [3]. Commercial instruments have been used frequently although home-made prototypes are also described in the literature [4].

In our work, a photodiode array spectrophotometer was mounted for use in flow injection analysis systems, using a monochromator in conjunction with a silicon photodiode array detector placed on the focal plane. For this implementation it was necessary to develop a computer program to control the data acquisition for each photodiode and to handle the data from the FIA manifold.

The computer program was developed in Visual Basic version 3.0, a programming language for the Windows environment that has been widely accepted by chemists [5,6]. Actually, Visual Basic does not support intrinsic functions capable of writing a byte into a specific microcomputer output device address, or of reading a byte from a microcomputer input device address. These functions are necessary to control the data acquisition, and to overcome this serious Visual Basic limitation a DLL library called INPOUT.DLL written in Assembler was employed. This library is available as public domain software in the Internet sites that offer Windows applications (for example, the Indiana University Windows repository site: ftp.cica.indiana.edu).

An important characteristic of Visual Basic is related to the multi-tasking operating environment of Windows. If Visual Basic and another application are running at the same time, automatic data exchange between programs is possible using dynamic data exchange (DDE). By using this fea-

*Corresponding author.

ture it is possible to improve the Visual Basic programs, using other Windows applications that also support DDE. The new versions of the Matlab for Windows (version 4.2 or later) [7] support the DDE conversation, making it possible to implement in the Visual Basic program many facilities supported by Matlab.

Matlab is a technical computing environment for high-performance numeric computation and data visualization, and its use has become popular in analytical chemistry, chemometrics, and process control research [8–10]. Matlab integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment that allows the user to do many matrix computations employing language commands that look very much like standard linear algebraic notation.

In the program developed, Visual Basic routines control the data acquisition and uses DDE to send data to Matlab for calculations and graphical visualization.

2. DDE conversation

Two Windows applications that support DDE can communicate by establishing a *DDE conversation*. The application that initiates the conversation is normally called the *client*, and the application that responds to the client is called the *server*. When a client application initiates a DDE conversation it must identify two DDE parameters: the name of the application it intends to have a conversation with, and the subject of the conversation, called the *topic*. During a DDE conversation, the client and server applications exchange data concerning *items*. An item is a piece of data actually being transferred during the DDE conversation.

The DDE application is performed by a Visual Basic program acting as client and Matlab as a server. Because Matlab acts as a server it is necessary to specify the service name, topic and item in the Visual Basic program. The service name must be 'matlab', and the topic could be 'System' or

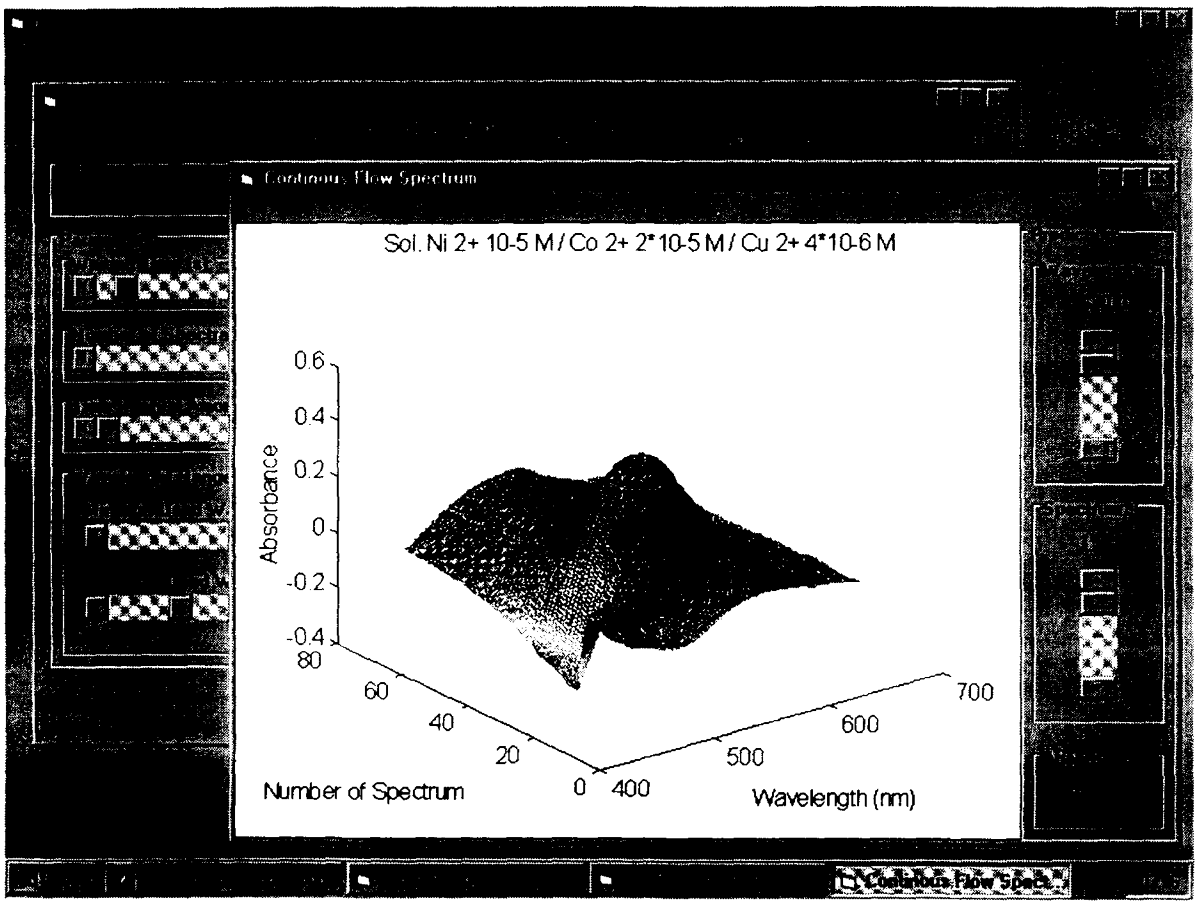

Fig. 1. Surface of a flow-injection system experiment generated in Matlab and transferred to a Visual Basic program via DDE.

Table 1
Visual Basic listing code for sending data to a specific microcomputer I/O address and for reading a analogue voltage data

```
Declare Function Inp Lib "inpout.dll" (ByVal Port%) As Integer
Declare Sub Out Lib "inpout.dll" (ByVal Port%, ByVal Value%)

Sub DigOut (Data)
'Subroutine to send a digital data to heximal address 300
   Out&H303,80 'all digital lines for output
   Out&H300,Data 'send data to a heximal address 300
End Sub

Sub Analogin (Data)
'Subroutine to read an analogue voltage with 12 bits precision
   Out&H310, 8 'sampling
   Out&H310, 0 'holding
   Out&H30C, 0 'start conversion
   For K%=1 to 10: next 'conversion time
   a%=Inp( and H30C) 'most significant bytes
   b%=Inp( and H30D) 'less significant bytes
   Value=((a%*16)+(b%/16))*10/4095 'tension value in voltage
End Sub
```

'Engine'. The 'System' topic allows users to browse the list of topics and items provided by the server, such as the formats supported by the server and not used in the current program. The Matlab 'Engine' topic was used, and permits three operations that may be used in a DDE conversation. The operations include sending commands for Matlab evaluation, requesting data from Matlab, and sending data directly to Matlab.

For sending commands to Matlab for evaluation, the item name required must be 'EngEvalString', and the command must be in text format. Visual Basic requests data from Matlab using the DDE request operation. It is possible to request strings using the 'EngStringResult' item, or request a graphic using the 'EngFigureResult'. The 'EngFigure Result' item can be used with text or Metafile pict formats. Visual Basic sends data to Matlab using the DDE poke operation. The 'Engine' topic supports DDE poke for updating or creating new matrices in the Matlab workspace. The item to be defined is the name of the matrix that will be sent. The matrix data can be in a text or XL Table format, a standard format of Microsoft Excel.

3. Home-made photodiode array spectrophotometer

An Oriel MultiSpec spectrograph with a crossed

Table 2
Visual Basic listing code of a DDE conversation to display a graphic generated by MATLAB

```
Sub Form_Graph1 ()
   TxtInput.LinkMode=none 'disable DDE (initiate conversation)
   TxtInput.LinkTopic="matlab|engine" 'service and topic name
   TxtInput.LinkItem="EngEvalString" 'item will be string
   TxtInput.LinkTimeout=1000 'timeout
   TxtInput.LinkMode=Link_manual 'operation mode
   TxtInput.LinkExecute="showspec" 'run function SHOWSPEC.M
   TxtInput.LinkMode=none 'disable DDE
   PicOutput.LinkMode=none 'enable new DDE operation
   PicOutput.LinkTopic="matlab|engine" 'service and topic name
   PicOutput.LinkItem="EngFigureResult" 'item will be figure
   PicOutput.LinkTimeout=1000 'timeout
   PicOutput.LinkRequest 'receive graphic generated in MATLAB
   PicOutput.LinkMode=none 'disable DDE
End Sub
```

Czern–Turner design that has been created specifically for photodiode array detectors, and the EG&G Reticon S-series photodiode array sensor with 1024 elements were used. The photodiode array sensor needs two electronic boards (RC 1000 and RC 1001) to provide all of the necessary digital signals and the video processing circuit to extract data. External 'start' and 'clock' digital inputs are provided at the edge connector of the motherboard (RC 1000) to allow control of the board operation by an external microcomputer. A 'start' pulse is necessary to initiate data acquisition, and 1024 clock pulses are sent to capture the output voltage values associated with each photodiode.

A Microquimica (SC-Brazil) interface board MQI 12PCC, containing a 12-bit analogue/digital converter and 24 digital programmable input/output communication lines was employed to provide external digital signals ('start' and 'clock' control signals) and for data acquisition. This interface board was plugged into an expansion slot of a IBM-PC type microcomputer, 486 DX/2, 66 MHz, with 8 Mb RAM.

4. The program to control the photodiode array

The process for spectrum acquisition requires first the capture of data corresponding to the voltage values of photodiodes. Visual Basic routines were implemented to read each photodiode voltage value, and to provide 'start' and 'clock' pulses from the microcomputer to the satellite board of the diode array. Routines for sending digital data to a specific microcomputer I/O address and for reading analogue voltage data, using the INPOUT.DLL library, are presented in Table 1.

After data acquisition, DDE communication is performed in which the data generated by the photodiode readings in the Visual Basic routines are transferred to Matlab for display of the spectrum. In the DDE conversation Visual Basic orders Matlab to execute a function called 'SHOWSPEC.M'. This function, stored in the Matlab workspace, loads the data generated by the photodiode readings and saved by Visual Basic in a microcomputer RAM drive. The graphic generated by Matlab (spectrum or surface) is transferred to Visual Basic, and a new window is opened to display it. Fig. 1 shows the window opened to display a surface obtained following the reaction of mixtures of Ni^{II}, Co^{II} and Cu^{II} with PAR [4-(2-pyridylazo) resorcinol] in a pH gradient generated by an FIA system. Seventy spectra in the range 430–630 nm were acquired at time intervals of one second.

The Visual Basic program code that performed the described DDE application is shown in Table 2, in a sub-program denoted 'Form_graph1'. The comments in each listed program line present all steps necessary to perform the DDE conversation. The program we describe is being updated for performing automatic multivariate calibration using the 'Chemometrics Toolbox' available in Mathworks.

Acknowledgements

The authors are grateful to the FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) for financial support (Process no. 94/2477-6), and Dr. Carol H. Collins for manuscript review.

References

[1] S.A. Borman, Anal. Chem., 55 (1983) 836A.
[2] L. Huber and S.A. George, Diode Array Detection in HPLC, Marcel Dekker, New York, 1993.
[3] M. Blanco, J. Gene, H. Iturriaga, S. Maspoch and J. Riba, Talanta, 34 (1987) 987.
[4] I.M. Raimundo Jr. and C. Pasquini, J. Autom. Chem., 15 (1993) 227.
[5] H.R. Keller, P. Fernandes de Aguiar and D.L. Massart, Trends Anal. Chem., 11 (1992) 131.
[6] R.G. Brereton, Analyst (Cambridge, UK), 119 (1994) 2149.
[7] MATLAB, MathWorks Inc., Sherborn, MA, USA.
[8] T.C. O'Haver, Chemom. Intell. Lab. Syst., 6 (1989) 95.
[9] B.M. Wise, Process Control Qual., 5 (1993) 73.
[10] S. Mitra and T. Bose, Chemom. Intell. Lab. Syst., 22 (1994) 3.

The authors are at the Instituto de Quimica, Universidade Estadual de Campinas, Caixa Postal 6154, 13083-970, Campinas, Brazil. E-mail: ronei@iqm.unicamp.br.